Talking to the Stars – Part II

Building an Interplanetary Internet based on Delay-Tolerant Network Protocols

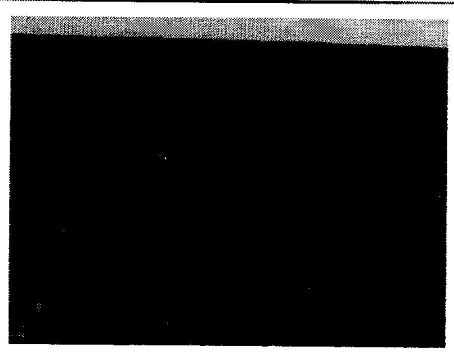
Scott Burleigh
NASA Jet Propulsion Laboratory,
California Institute of Technology
19 January 2004

A Typical Interplanetary Internet Node

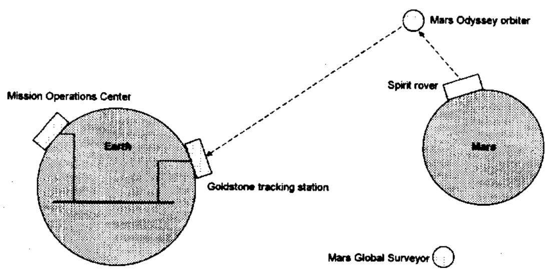
First image of the newly-landed "Spirit" Mars Exploration Rover

An Image Transferred from Spirit

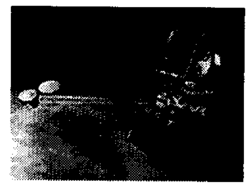
View of the Martian Horizon



How Spirit Sent Those Images From Mars



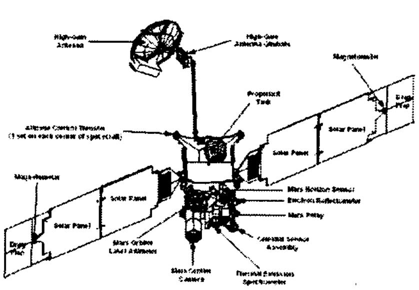
Orbiters Used for Data Relay from Mars



Mars Odyssey

Mars Global Surveyor

Mars Global Surveyor



So, This Is Easy, Right?

- Actually it's harder than it looks, even for the simple case of one rover (soon to be two, we hope) and two orbiters.
- Structure of this talk:
  - What makes deep-space communications hard?
  - How have we done it in the past?
  - Why isn't that good enough for the future?
  - Why not just use Internet protocols?
  - What are we using today instead?
  - Why isn't that good enough for the future?
  - What can we do instead?
  - How far away is that stuff?
  - I'll stop asking myself easy questions; your turn.

Deep-Space Communication Challenges

- Spacecraft have limited power and antenna size, so data rates are typically low and are often highly asymmetrical.
- Links are noisy due to solar wind, etc.
- But the central problem is extremely long round-trip communication times:
  - The distances are very long, and the speed of light is fixed, so signal propagation delay is on the order of minutes or hours rather than milliseconds.
  - Connectivity is intermittent. For example, the Deep Space Network antenna complexes (Goldstone, Madrid, Canberra) may "track" a given spacecraft for only 2 hours per day – or only 8 hours once per week. If a transmission reaches a spacecraft at the end of a tracking pass, the response can't be received until the start of the next pass.

The Reliability Problem

- We use a lot of forward error correction coding in transmissions from spacecraft, but this doesn't assure perfect communication. Data lost in transit still need to be retransmitted somehow.
- Because round-trip times can be very long, the reliable transmission of any single byte can theoretically take an arbitrarily long time:
  - Transmission can be lost due to corruption, N times.
  - NAK (retransmission request) can itself be lost due to corruption, N times.
  - Connectivity can be lost between time of transmission and time of reception, so transmission of NAK (or of data) in response can be delayed by hours or days.

Timing is Everything

- To get data flowing from A to B at time T, where A and B are deep-space entities (e.g., spacecraft and ground station):
  1. At time T - x, where x is however long it will take to point A's antenna, the computer for the device (spacecraft, rover, ground station) that A resides in has to start moving A's antenna so that it points at wherever B will be at time T.
     - x may be substantial: articulated antennae may re-point fairly quickly, but pointing the body-fixed high-gain antenna that many smaller spacecraft carry will entail re-orienting the entire spacecraft.
     - This itself may be non-trivial: for example, if the rotation of the spacecraft will cause the star scanner to be on the sunward side of the spacecraft at some point, you have to make sure that the shutter over the star scanner will be closed or you'll burn out the star scanner.

Timing is Everything (cont'd)

2. At time T or somewhat before, power has to be applied to A's transmitter.
   - Spacecraft in cruise with plenty of solar power may be able to power their radios continuously, but highly power-constrained devices like rovers need to be careful not to waste precious electricity.
   - It may be necessary to apply power to a heater – to warm up the transmitter to a temperature at which it can operate – some number of minutes before powering the radio itself.
3. At time T, A starts radiating data to B on whatever frequency it knows B will be listening on at time T2.
   - Time T2 is equal to T + L, where L is the distance that B will be from A at time T expressed in light seconds.
4. At time T2 - y, where y is however long it will take to point B's antenna, the computer for the device that B resides in has to start moving B's antenna so that it points at wherever A was (or will be, depending) at time T. Again, y may be substantial.

Timing is Everything (cont'd)

5. At time T2 or somewhat before, power has to be applied to B's receiver.
   - Again power management may be important.
   - In fact, the power consumed by a receiver may be greater than that consumed by the same radio's transmitter.
6. At time T2, B will start receiving the first bits radiated by A, on whatever frequency it knows A was transmitting on at time T.

- Summing up: none of this happens spontaneously. Without exhaustive planning, coordinated schedules, and synchronized clocks, communication opportunities are lost.

How We've Done this in the Past

- Before the 1990s, the whole answer was radio engineering and manual operations.
  - Uplink (telecommand) regarded as wholly distinct from downlink (telemetry).
  - "Discrete" radio signals for simple, direct commanding of spacecraft hardware.
  - All onboard memory managed from the ground; memory uploads to revise flight software.
  - All telemetry was time-division-multiplexed, commutated and decommutated.

CCSDS in the '90s

- CCSDS (Consultative Committee for Space Data Systems) protocols standardized this model:
  - Recommendations at physical, link, and network layers of the protocol stack, to enable interoperation of different national space agencies' spacecraft and ground stations.
  - Telemetry and telecommand data bundled in CCSDS *packets*.
  - Standards for framing at link layer, for coding, and for waveforms.
  - Forward error encoding on downlink: Reed-Solomon.
  - Optional link-layer ARQ on uplink: automatic retransmission of telecommand frames on "go back N" model.
    - Command link control words are inserted into telemetry frames.
- Standards very broadly adopted, used on hundreds of spacecraft.

But the Low-level Protocols Aren't Enough

- Still labor-intensive, so operations costs remain relatively high.
- No standard automated systems for retransmission.
  - In the late '90s, automatic reliability systems began to be built on the CCSDS protocols to reduce costs.
    - Telemetry packet retransmission systems on downlink for Mars Pathfinder, DS-1, SIRTF.
    - Content-independent uplink protocol (CIUP) for the DS-1 spacecraft.
  - But these systems had limited functionality and were not standardized.
    - Not useful for cross-support between different space agencies' spacecraft and ground tracking networks.
- No automated systems for relay operations.
  - All MER relaying is manually planned and commanded.

TCP/IP Works Great, But Not for Deep Space

- Long round-trip times constrain reliable transmission protocols:
  - Connection establishment could take days.
    - So protocol must be connectionless.
  - In-order stream delivery could suffer arbitrarily long periods of paralysis, waiting for byte N to be received before delivering byte N + 1.
    - So out-of-order delivery is needed.
    - So protocol must support multiple transmissions in flight concurrently.
    - So data must be structured in self-identifying messages (transmission blocks) for accountability and concurrent retransmission; not in streams.
  - Any single message transmission can take an arbitrarily long time.
    - So any number of message transmissions might be in progress at the moment a computer is rebooted or power cycled.
    - So retransmission buffers should reside in non-volatile storage to minimize risk of massive transmission failure.

17 19 January 2004

More Constraints

- Continuous end-to-end transmission through relay elements may be impossible, due to time-disjoint episodes of connectivity.
  - So relays can't just route packets; they must store them, and then forward them when opportunities arise.
- End-to-end retransmission would reserve resources (retransmission buffer) at originator for entire duration of the transaction – possibly days or weeks.
  - So retransmission should be point-to-point rather than end-to-end. "Custody transfer."

18 19 January 2004

TCP Issues

- TCP **connection establishment** can consume an excessive fraction of access opportunities, especially when signal propagation latency is high and access opportunity is brief (or data rate is low).
- TCP's **in-order data delivery** imposes a round-trip-time delay on data arrival at the application whenever there is any data loss.
- TCP's **congestion control** response to data loss severely limits throughput when signal propagation latencies are high.
- **End-to-end retransmission** in TCP consumes excessive storage at data sources with limited resources (e.g., instruments) when round-trip-time delays are high.

19 19 January 2004

IP Routing Issues

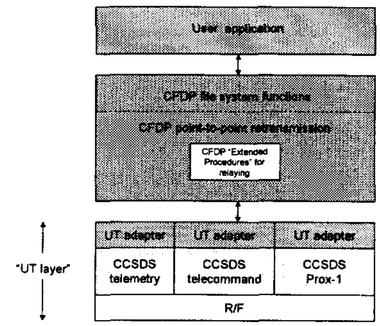
- Border Gateway Protocol (BGP) **runs over TCP**, subject to the same problems.
- Local routing protocols operating within Autonomous Systems respond poorly to intermittent connectivity.
  - They rely on **periodic reachability reports** from agents.
  - Transient network partitioning can interrupt this reporting and be interpreted as sustained loss of reachability on a network link.
  - A series of these losses, concatenated, can be misinterpreted as loss of end-to-end reachability.

20 19 January 2004

CFDP (CCSDS File Delivery Protocol)

- A single international standard for automatic, reliable transfer of files between spacecraft and ground (in both directions) over interplanetary distances, built on the CCSDS low-level protocols.
- Monolithic – a single protocol that performs:
  - File transfer and remote file system management, over...
  - (optional) end-to-end relaying through a simple network, over...
  - (optional) point-to-point retransmission for end-to-end reliability.
- Relay features:
  - Deferred transmission: source node retains file data in persistent storage until contact with initial relay is established.
  - Store-and-forward operation: relay node retains file data in persistent storage until contact with next relay – or destination node – is established.

CFDP Architecture



The Wave of the Near Future

- First use of CFDP in flight was in late 2002, on AlSat-1 (Algerian observation satellite built by Surrey Space Technology, Ltd.).
- Upcoming CFDP deployments:
  - MESSENGER mission to Mercury, from Johns Hopkins Applied Physics Laboratory, launching in May of 2004.
  - Deep Impact comet investigation mission, from JPL, launching in December 2004 or January 2005.
- CFDP is also baselined for Mars Reconnaissance Orbiter and the proposed Mars Science Laboratory, and it's under study for other new JPL missions.

But CFDP is Also Limited

- No routing protocol.
  - Route computation is performed using static routing tables.
  - Routing table modification must be performed by flight software external to CFDP, e.g., under mission operations command.
- No support for reliable relay through multiple parallel relay nodes.
  - When a file is too large to relay in a single contact period, we can either wait for the next contact with the same relay node or else relay part of the file through the next contact with a different relay node – which may be sooner. So parallel relay can accelerate the release of resources at the source node.
  - But all CFDP reliability protocol interchange for a single point-to-point transfer of a given file must be conducted between the same two nodes.
  - So for parallel relaying, file must be divided into multiple partial files that can be serially relayed, then reconstituted at the final destination. CFDP provides no standard support for file division and reconstitution.

Suppose Things Go Well

- CFDP relaying is enough for networks of relatively static topology and small size, where file transmission sizes match contact durations. That's what it was designed for.
- But early and continued success in robotic Mars exploration might result in a more intensive program. Could CFDP run a sensor network?

Managing a Mars System Enterprise

- Changes:
  - Number of nodes increases by orders of magnitude.
  - Number of possible node interconnections increases exponentially.
  - Number of routing options and relaying opportunities increases as a function of topological complexity.
  - Intermediate relaying opportunities offered by mobile or dual-use nodes are often brief and may be wholly opportunistic.
- Implications:
  - Due to growth in the number of possible routes, routing must be dynamic and rapidly responsive to changing local conditions.
  - Due to growth in the number of short-duration relay opportunities, files must be dynamically divided for partial transmission on parallel routes.
- This dynamic behavior is not built into CFDP, and accomplishing it by command from Earth over round-trip times of 8 to 40 minutes would be difficult.

Delay-Tolerant Networking (DTN)

- DTN features overview:
  - *Bundling* protocol builds on and includes all of the concepts built into CFDP relaying: deferred transmission, store-and-forward operation, underlying point-to-point retransmission for end-to-end reliability.
  - Adds automatic dynamic route computation, adapted from routing experience in the Internet.
  - Adds automatic reactive fragmentation to deal with truncated contacts.
  - Also adds built-in support for security and congestion avoidance.
- Unlike CFDP relaying, delay-tolerant networking architecture is designed to scale up indefinitely.
- Unlike CFDP, DTN is not deep-space-specific but is designed for seamless integration with the Internet. Conceptually, scientist on workstation at home institution interacts directly with instrument on spacecraft 20 light minutes away.

Architectural Overview

- *Overlay network* operational objectives:
  - Run over Internet protocols wherever possible.
  - Run over domain-specific (e.g., CCSDS) protocols as necessary.
  - Insulate applications from having to know the difference.
- DTN design principles:
  - A **postal model** of communications.
    - Telephonic, conversational communication is a special case that only works under favorable conditions. Epistolary communication is the more general and more robust model.
    - Forego dialogue and negotiation; instead, "bundle" with each message the answers to questions that might be asked about it.
  - Tiered functionality.
    - Use overlay network protocol to do whatever the underlying transmission systems cannot, but no more.
  - Terseness.

Least-common-denominator Transmission

- DTN end-to-end overlay network architecture must span both Internet-like and high-latency environments, tolerating the deficiencies of both.
- Design cannot rely on any end-to-end expectation of:
  - continuous connectivity anywhere in the network
  - low or constant signal propagation latency
  - low error rate
  - low congestion
  - high transmission rate
  - symmetrical data rates (transmission and reception)
  - data arrival in transmission order
  - common name or address expression syntax or semantics

DTN Design Elements (1 of 3)

- Tiered forwarding:
  - Underlying network protocols (such as IP) are invoked wherever possible; Bundling need not be invoked at every IP router.
  - The Bundling overlay network protocol operates at sub-network boundaries where the underlying network protocols must terminate.
- Deferred transmission: store bundles locally, within the network, until the next forwarding opportunity arises.
- Tiered routing:
  - Underlying networks' routing protocols support the underlying network protocols.
  - DTN routing is based on awareness of forwarding opportunities (*contacts*), which may be continuous, on-demand, scheduled, predicted, or opportunistic.

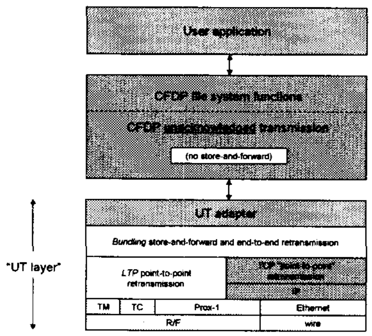
DTN Design Elements (2 of 3)

- Tiered ARQ:
  - Performed by underlying transport systems (e.g., TCP) where supported.
  - Optional *custody transfer* retransmission supported by Bundling.
- Tiered security:
  - Bundling infrastructure protected by bundle sender authentication.
  - End-to-end data integrity and confidentially service may also be provided by Bundling; no firm design decision yet.
- Tiered congestion avoidance:
  - Congestion avoidance in underlying transport systems is assumed.
  - Bundling responds to congestion in the overlay network by invoking tiered flow control.

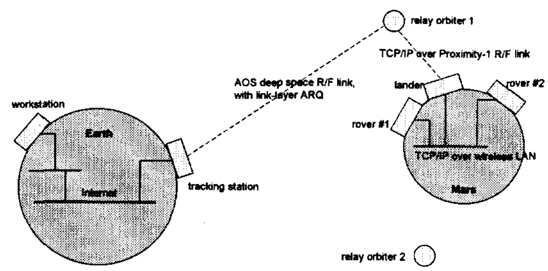
DTN Design Elements (3 of 3)

- Tiered flow control:
  - Flow control in underlying transport systems may be protocol-based (as in Internet) or managed, rate-based (as on deep space links).
  - Bundling invokes underlying flow control systems by refusing to accept custody of bundles.
- Tiered coding:
  - Forward error correction as needed in underlying transport systems.
  - Optional additional coding in Bundling for header compression.
- Tiered fragmentation and reassembly:
  - Performed by underlying network protocols as required by links.
  - Performed by Bundling as required by contact intermittency.
- Resilient delivery: deferred delivery, destination reanimation.
- Postal service levels: priorities, notification services.

CFDP/DTN Architecture



An Application Example



DTN Development Status

- Summer of 2002: first demonstration of reference implementation of Bundling.
- March 2003: peer review of DTN architecture Internet Draft #2.
- Late spring 2003: Bundling reference implementation released for open source evaluation and contribution.
- November 2003: JPL implementation of Bundling, designed for qualification as flight software, begins initial testing.
- December 2003: DARPA announces Proposer's Day on 21 January 2004 for its DTN program "new start". BAA (Broad Agency Announcement) date is TBD.
- January 2004: Mars Telecom Orbiter pre-project begins evaluating DTN. Proposed MTO launch date is 2009.
- But protocol specification not yet frozen; formal standardization efforts (within either IETF or CCSDS) have not yet begun.

To Find Out More...

- www.dtnrg.org is the web site for the DTN Research Group of IRTF (the Internet Research Task Force).
- www.ipnsig.org is the web site for the InterPlanetary Internet Special Interest Group of the Internet Society.
- www.ccsds.org is the web site for the Consultative Committee for Space Data Systems.